RDC

ROCKWELL DESIGN CENTER

LOW COST EMULATOR (LCE)

PROM PROGRAMMER MODULE USER'S MANUAL

Rockwell International

Semiconductor Products Division

©Rockwell International Corporation 1985
All Rights Reserved
Printed in U.S.A.

Document No. 29655N25
Order No. RDC25
March 1985

RDC

ROCKWELL DESIGN CENTER

LOW COST EMULATOR (LCE)

PROM PROGRAMMER MODULE USER'S MANUAL

Rockwell International

Semiconductor Products Division

©Rockwell International Corporation 1985
All Rights Reserved
Printed in U.S.A.

Document No. 29655N25
Order No. RDC25
March 1985

NOTICE

Rockwell International does not assume any liability arising out of the application or use of any products, circuit or software described herein, neither does it convey any license under its patent rights nor the patent rights of others. Rockwell International further reserves the right to make changes in any products described herein without notice. This document is subject to change without notice.

WARNING

This equipment generates, uses and can radiate radio frequency energy and if not installed and used in accordance with the instruction manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

INFORMATION TO USER

If this equipment does cause interference to radio or television reception which can be determined by turning the equipment on and off, the user is encouraged to try to correct the interference by one or more of the following measures:

- reorient the receiving antenna
- relocate the computer with respect to the receiver
- move the computer away from the receiver
- plug the computer into a different outlet so that the computer and receiver are on different circuits.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet prepared by the Federal Communications Commission helpful:

"How to Identify and Resolve Radio and TV Interference Problems"

This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20402, Stock Number 004-000-0345-4.

TABLE OF CONTENTS

Section		Page
	SECTION 1	
	INTRODUCTION	
1.1	Overview	1-1
1.2	Features	1-1
1.3	Characteristics	1-3
1.4	Related Documents	1-3
	SECTION 2	
	INSTALLATION	
2.1	Unpacking	2-1
2.2	Operating Options	2-1
	2.2.1 Base Address Selection	2-4
	2.2.2 Bank Selection	2-4
	2.2.3 Firmware ROM Range Selection Jumpers	2-5
2.3	Installation and Removal	2-6
	2.3.1 Module Connection and Setup	2-6
	2.3.2 Module Removal	2-8
2.4	PROM Installation and Removal	2-8
	2.4.1 PROM Installation	2-9
	2.4.2 PROM Removal	2-9
	SECTION 3	
	STARTUP AND OPERATION OVERVIEW	
3.1	General Operation	3-1
3.2	PROM Programmer Initialization	3-2
3.3	PROM Programmer Modes	3-4
3.4	General System Characteristics	3-7

Section Page

3.4.1 Softkey Selection 3-7
3.4.2 ESCAPE, MINUS and ZERO keys 3-7
3.4.3 Address Entry 3-7
3.4.4 Command Verification 3-8
3.4.5 In Process Indication 3-8
3.4.6 Improper PROM Insertion 3-9

SECTION 4
PROM PROGRAMMER MEMORY COMMANDS

4.1 DUMP 4-2
4.1.1 Standard Dump 4-2
4.1.2 Single Byte Option 4-3
4.2 MODIFY 4-4
4.3 ALTER 4-5
4.4 INVERT 4-6
4.5 SAVE 4-7
4.6 LOAD 4-8
4.7 VERIFY 4-9

SECTION 5
PROM PROGRAMMER PROM COMMANDS

5.1 CHECK 5-1
5.2 PROGRAM 5-3
5.3 READ 5-8
5.4 COMPARE 5-9
5.5 ERASE 5-11
5.6 Copying a PROM 5-12

Section Page

SECTION 6
PROM PROGRAMMER UTILITY COMMANDS

6.1 CHEKSUM 6-1
6.2 TOG VER 6-3
6.3 SELECT 6-3

SECTION 7
FUNCTIONAL AND INTERFACE DESCRIPTIONS

7.1 Memory Map 7-1
7.2 Functional Description. 7-3
7.3 Interface Description. 7-8

APPENDICES

Appendix Page

A Schematic Diagrams A-1

LIST OF FIGURES

Figure		Page
1-1	RDC-3030 PROM Programmer Module	1-2
2-1	LCE PROM Programmer Module Details	2-2
2-2	PROM Programmer Module Location Within the LCE	2-7
2-3	LCE Front Panel	2-8
3-1	PROM Concept Example	3-6
7-1	LCE Memory Banking Map	7-2
7-2	PROM Programmer Module Block Diagram	7-4

LIST OF TABLES

Table		Page
1-1	EPROM and EEPROM Devices Supported	1-4
1-2	PROM Programmer Module Physical and Electrical Characteristics	1-5
2-1	PROM Programmer Module Switch, Jumper, Connector and Potentiometer Descriptions	2-3
2-2	Base Address Selection Switch Positions	2-4
2-3	Bank Selection Switch Positions	2-4
2-4	Firmware ROM Selection Jumper Position	2-5
3-1	PROM Programmer Commands	3-5
7-1	PROM Programmer Module Memory Map	7-1
7-2	Output Voltage Selections	7-8
7-3	Connector P1 (RM 65 Bus) Pin Assignments	7-9
7-4	Connector P1 (RM 65 Bus) Signal Descriptions	7-10
7-5	Connector J1 (PROM Interface) Pin Assignments	7-11
7-6	Connector J1 (PROM Interface) Signal Description	7-11

SECTION 1

INTRODUCTION

1.1 OVERVIEW

In conjunction with the Low Cost Emulator (LCE), the RDC-3030 LCE PROM Programmer Module (Figure 1-1) can be used to program UV EPROM (ultra-violet light erasable programmable read-only memory) and EEPROM (electrically erasable programmable read-only memory) devices. Both devices are generically referred to as PROMS. For EPROMs, capabilities provided are: programming either a complete PROM or any portion of the PROM, checking the PROM (to determine if it is initialized), reading the PROM, verifying the PROM, altering memory, filling unused memory locations, inverting memory contents, exiting to the LCE PRIMARY menu level, and performing other utility-type functions. For EEPROMs, the same functions are provided as well as the erasing capability.

1.2 FEATURES

- o RDC RM 65 Bus compatible
- o Supports industry-standard 2K, 4K, 8K, and 16K EPROMs and 2K byte EEPROMs (Table 1-1)
- o Stricty controlled programming voltage rise and fall times ensure compatibility with many devices and prevent overshoot
- o DAC controlled programming voltage provides compatability with present and future technologies
- o Connects to LCE front panel-mounted 28-pin Zero Insertion Force (ZIF) PROM socket
- o Firmware ROM is jumper selectable to operate with a user-provided 8K or 4K ROM, or disables firmware for I/O operation only
- o Base Address switches to select the PROM Programmer firmware and I/O registers for operation with common 64K-byte memory banks
- o On-board DC-to-DC converter generates the applicable programming voltage, yet allows 5 Vdc only operation.
- o Fully assembled, tested, and warranted

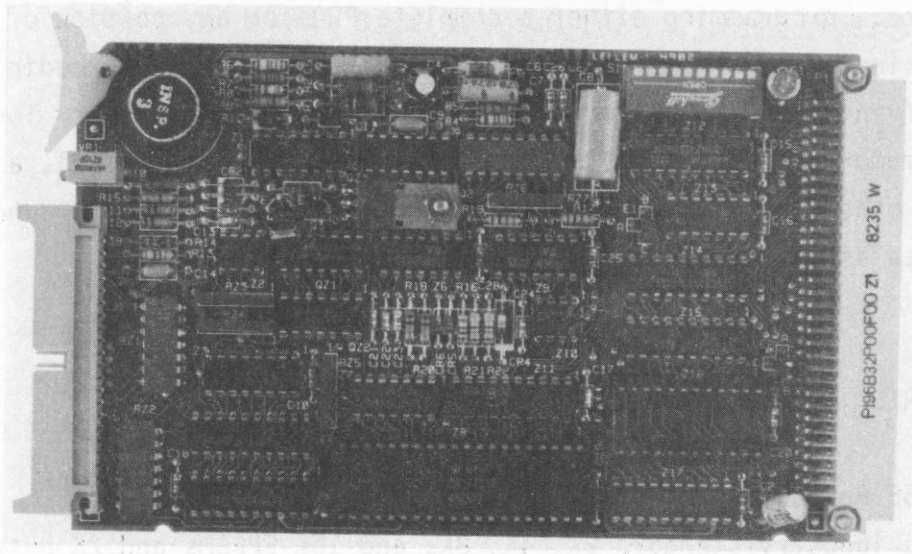


Figure 1-1. RDC-3030 LCE PROM Programmer Module

1.3 CHARACTERISTICS

Physical and electrical characteristics of the LCE PROM Programmer Module are listed in Table 1-2. Outline details are shown in Figure 1-2.

1.4 RELATED DOCUMENTS

The following Rockwell documents provide additional information about the LCE installation and operation.

Order No./Document No.	Title
RDC19/29655N19	Low Cost Emulator (LCE) System User's Manual
RDC20/29655N20	Low Cost Emulator (LCE) Hardware Installation Manual
RDC16/29655N16	Target RAM Module Installation Manual

Table 1-1. EPROM and EEPROM Devices Supported

Device	Device Number	Manufacturer	Device Number Entered
EPROMs			
2K x 8	2516	Texas Instruments	2716
	2716	AMD, Fujitsu, Hitachi, Intel, Mostek, Motorola, National, NEC, Texas Instruments, Toshiba	2716
4K x 8	2532	Hitachi, Motorola, Texas Instruments	2532
	2732	AMD, Fujitsu, Hitachi, National, NEC, Toshiba	2732
	2732A	Intel	2732
	87C32	Ricoh	87C32
8K x 8	2564	Texas Instruments	2564
	2764	Fujitsu, Intel, SEEQ	2764
	68764	Motorola	68764
	68766	Motorola	68766
	87C64	Rockwell	87C64
16 K x 8	27128	Intel, SEEQ	27128
EEPROMs			
2K x 8	2816	Intel	2816
	5213	SEEQ	5213
	48016	Hitachi	48016

Table 1-2. PROM Programmer Module Physical and Electrical Characteristics

Characteristic	Value
Dimensions	
PROM Programmer Module	
Width	100 mm (3.94 in.)
Length	167 mm (6.58 in.)
Height (See Note 1)	14 mm (0.56 in.)
Weight	184 g (6.5 oz.)
PROM Programmer Cable Length	610 mm (24 in.)
Environment	
Operating Temperature	0° to 70°C
Storage Temperature	-40°C to 85°C
Relative Humidity	0% to 85% (without condensation)
Power Requirements	+5 Vdc ±5% at 1.1 A typical 2.0 A maximum (average) 2.9 A maximum (peak)
Connectors/Sockets	
RM 65 Bus Connector (P1)	64-pin plug per DIN 41612 (Rows a and b with c not installed)
Socket Module Cable Connector (J1)	40-pin plug (0.100 in. centers) per DIN 41612, mates with 3M 3417-7040 or equivalent
PROM Socket	28-pin Zero Insertion Force (ZIF)

NOTE

1. Height value includes the maximum values for component height above the board surface (0.4 in.) printed circuit board thickness (0.062 in.), and pin extension through the bottom of the module (0.1 in.)

SECTION 2

INSTALLATION

2.1 UNPACKING

Remove the module from the shipping carton and, referring to the packing sheet, verify that all parts listed have been shipped. Save the packing materials for storing the module.

CAUTION

> This module contains voltage-sensitive items. The module should be stored in an antistatic container when not in use, and anyone handling the unit should observe antistatic precautions. Damage to the unit can occur if antistatic protection is not maintained.

2.2 OPERATING OPTIONS

Options available at installation are:

- Base Address Selection
- Bank Selection
- Firmware ROM Selection Jumpers

Physical details of the PROM Programmer module are indicated in Figure 2-1. Functions of switches, jumpers and connectors shown in Figure 2-1 are listed in Table 2-1. Also listed in Table 2-1 are references to paragraphs and other tables containing additional information.

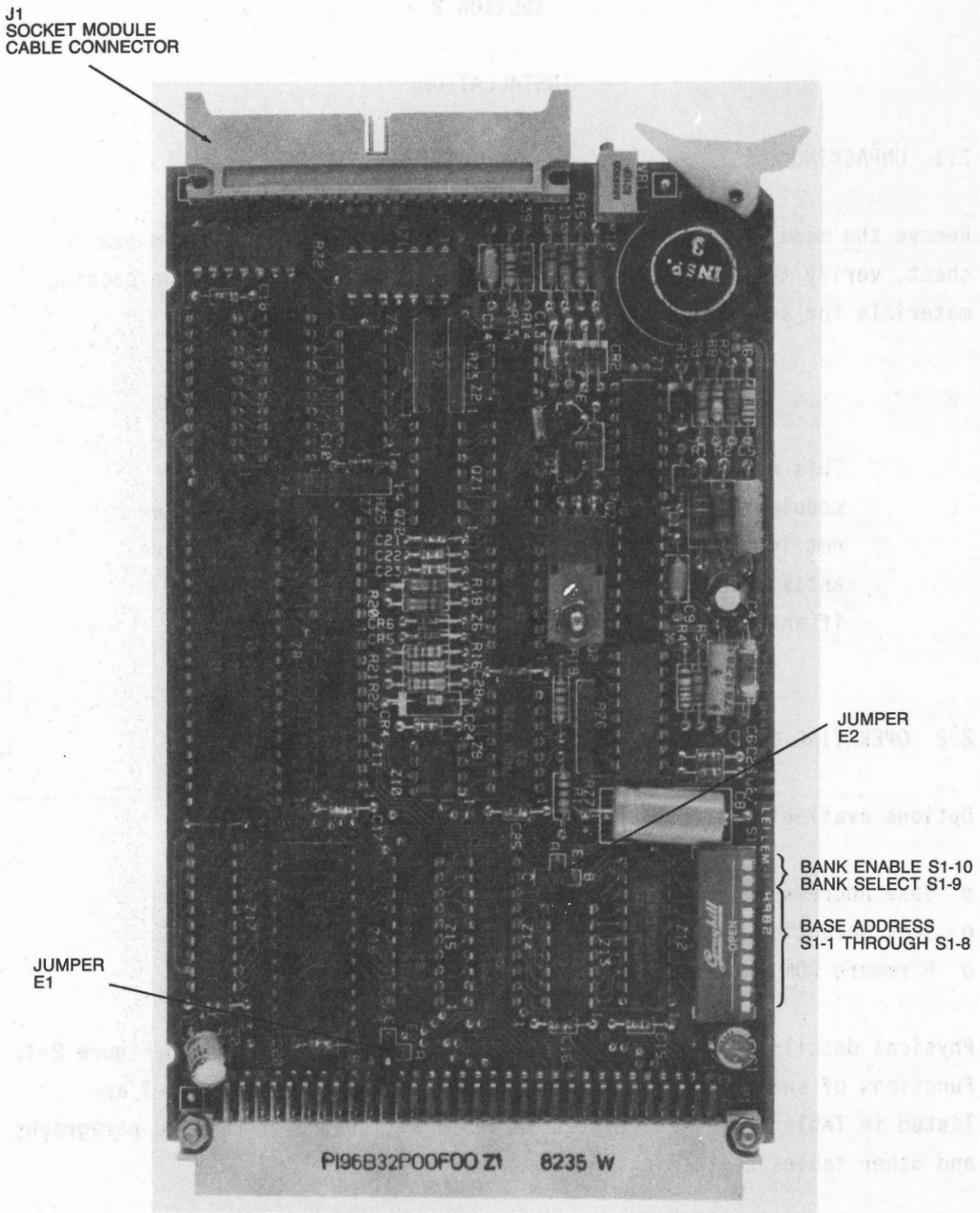


Figure 2-1. PROM Programmer Module Details

Table 2-1. PROM Programmer Module Switch, Jumper and Connector Descriptions

Category	Item	Description	Reference
Switches	S1-1 to S1-8	Base Address Switches	2.2.1 and Table 2-2
	S1-9	Bank Select Switches	2.2.2 and Table 2-3
	S1-10	Bank Enable Switches	2.2.2 and Table 2-3
Jumpers	E1	Firmware ROM Range Selection Jumper	2.2.3 and Table 2-4
	E2	Firmware ROM Type Selection Jumper	2.2.3 and Table 2-4
Connectors	P1	RM 65 Bus Connector	Tables 7-3 and 7-4
	J1	Socket Module Cable Connector	Tables 7-5 and 7-6

2.2.1 Base Address Selection

For LCE operation, the PROM Programmer Module firmware must be located at $A000-$BEFF and the module I/O must be located at $BF00-$BFFF (see Table 2-2). Set switches S1-1 through S1-4 to hex value XFXX to locate the I/O. Set switches S1-5 through S1-8 to hex value AXXX to locate the firmware.

Table 2-2. Base Address Selection Switch Positions

Switch S1-5 - S1-8 Positions					Switch S1-1 - S1-4 Positions				
Address (Hex)	S1-8	S1-7	S1-6	S1-5	Address (Hex)	S1-4	S1-3	S1-2	S1-1
AXXX	CLOSED	OPEN	CLOSED	-	XFXX	CLOSED	CLOSED	CLOSED	CLOSED
NOTE: S1-5 is ignored and can be in either position									

2.2.2 Bank Selection

For LCE operation, memory bank 0 operation must be selected (see Table 2-3). Set S1-10 to CLOSED to enable bank selection and S1-9 to OPEN to select bank 0.

Table 2-3. Bank Selection Switch Positions

Memory Bank Selected	Switch Position	
	Bank Enable Switch S1-10	Bank Select Switch S1-9
Bank 0 Only	CLOSED	OPEN

2.2.3 Firmware ROM Range Selection Jumpers

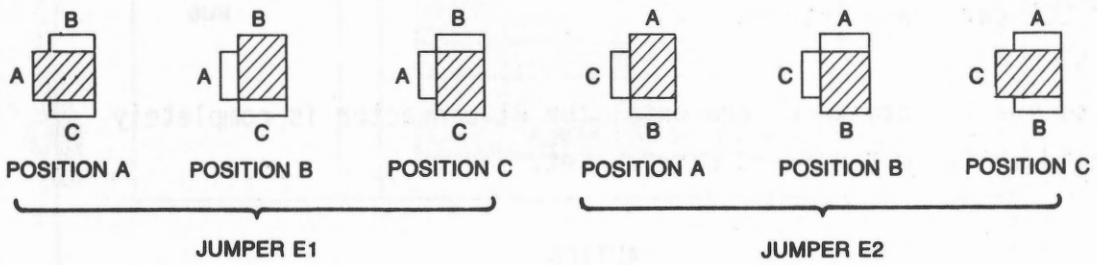
For LCE operation, jumpers E1 and E2 must be installed to select operation of a 2764 PROM containing the PROM Programmer firmware on-board the PROM Programmer module (see Table 2-4). Jumper E1 in position A selects on-board operation and jumper E2 in position C enables 2764 operation.

Table 2-4. Firmware ROM Selection Jumper Positions

Firmware Option	Jumper	Position
On-Board 2764 PROM	E1=A	E2=C

NOTE

Jumper positions with the module ejector tab located in the upper left corner are as follows:



2.3 INSTALLATION AND REMOVAL

2.3.1 Module Connection and Setup

To install the module:

a. Disconnect system power.

b. Verify that S1-1 through S1-8 switches are set correctly (Paragraph 2.2.1 and Table 2-2).

c. Verify that S1-9 and S1-10 Bank Selection switches are set correctly (Paragraph 2.2.2 and Table 2-3).

d. Install E1 and E2 jumpers as required (Paragraph 2.2.3 and Table 2-4).

e. Slide module into the available circuit board slot at the top left of the LCE card cage (Figure 2-2).

f. Press module into card cage until the P1 connector is completely seated in the mating card cage socket.

CAUTION

If the module cannot be inserted into the interfacing connector (receptacle) with moderate pressure applied, check the orientation and connector alignment of the module. Forcing the module improperly into the receptacle can damage the receptacle an/or the module.

g. Connect the cable from the PROM socket connector module to the PROM Programmer Module J1 connector so that arrow symbol on the module connector (pin 1 locator) aligns with the red strip on the cable (pin 1 locator).

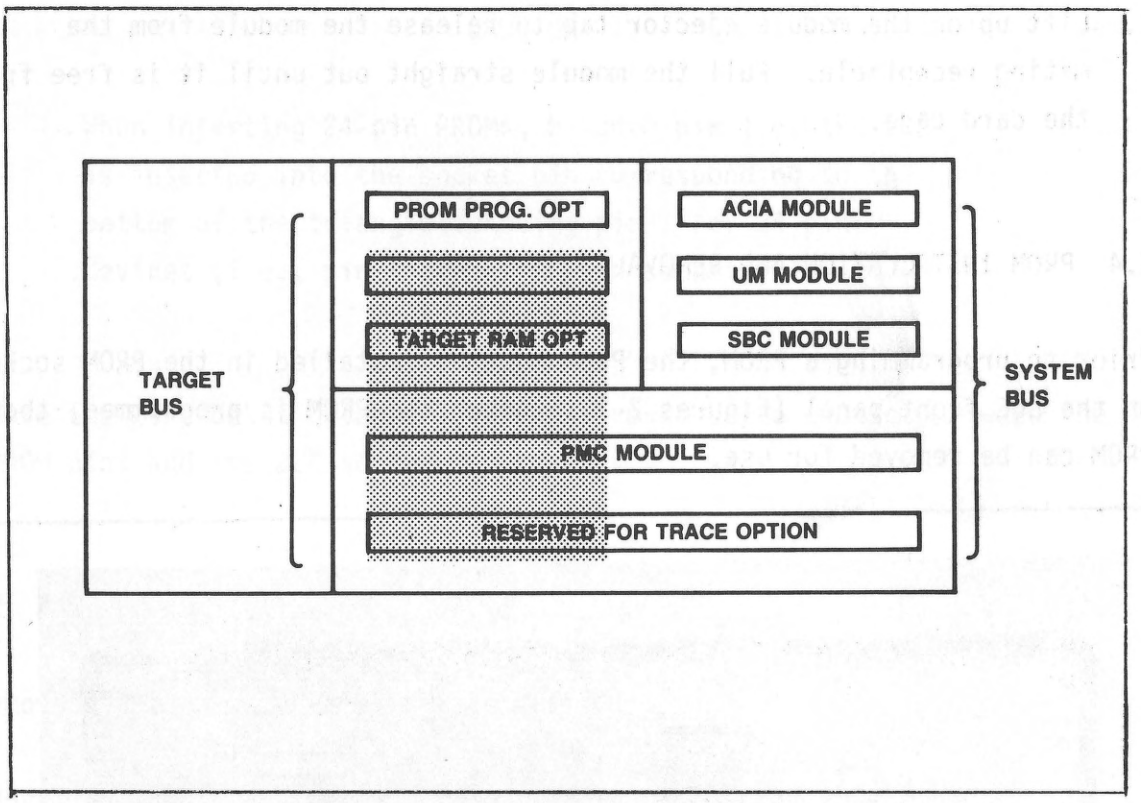


Figure 2-2. PROM Programmer Module Location Within the LCE

2.3.2 Module Removal

To remove the module:

a. Disconnect system power.

b. Disconnect PROM Socket module cable connector from PROM Programmer Module J1.

c. Lift up on the module ejector tab to release the module from the mating receptacle. Pull the module straight out until it is free from the card cage.

2.4 PROM INSTALLATION AND REMOVAL

Prior to programming a PROM, the PROM must be installed in the PROM socket on the LCE front panel (Figures 2-3). After the PROM is programmed, the PROM can be removed for use.

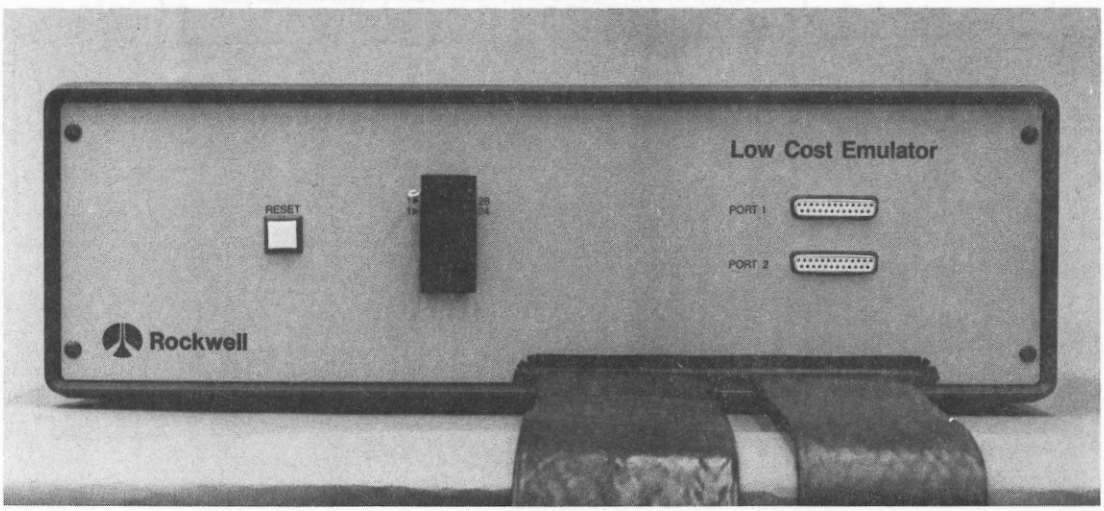


Figure 2-3. LCE Front Panel

2.4.1 PROM Installation

To install a PROM into the LCE PROM socket:

a. Move the ZIF socket release lever down to release the socket pin contacts.

b. Insert the PROM to be programmed into the ZIF Socket.

CAUTION

When inserting 24-pin PROMs, be sure pin 1 of the PROM is inserted into the socket pin corresponding to the bottom of the triangle denoting pin 1 for 24-pin devices (i.e., pin 3 of the 28-pin socket).

c. Move the ZIF socket release lever up to make solid contact between the PROM pins and the ZIF socket.

2.4.2 PROM Removal

To remove a PROM from the ZIF PROM socket:

a. Move the ZIF socket down to release pressure on the PROM pins.

b. Remove the PROM from the ZIF socket.

SECTION 3

STARTUP AND OPERATION OVERVIEW

3.1 GENERAL OPERATION

PROM Programmer Module initialization instructions and command operations are described in this section. Command descriptions are based on the premise that module initialization procedures have been completed.

NOTE

> Except where noted in this manual, PROM is used to mean either EPROM or EEPROM. Also, unless otherwise noted, all data and addresses entered in response to prompts are in hexadecimal.

Prior to programming a PROM, the PROM is usually initialized or erased (all bits set to "1's"). To initialize an EPROM, the EPROM must be exposed to ultraviolet light for the amount of time specified by the manufacturer. Usually, the erase time is from 2 to 20 minutes. However, an EEPROM can be initialized by using the ERASE command while using the PROM Programmer Module. During programming, selected bits in PROM locations are switched to "0's," however, once set to "0's", those bits cannot be reset to "1's" without reinitializing the PROM.

Enter a requested address as a four-digit number without a leading "$". Correct a typing error by using DEL or by entering four additional correct numbers. End all data inputs with RETURN or SPACE unless otherwise directed. Some commands are self-initiating upon entry of a code, and do not require pressing RETURN or SPACE. Terminate and reenter one of these commands if an entry error is made.

NOTES

1. A personality module must be connected to the LCE for the PROM Programmer Module to operate properly.

2. An 8K, 16K or 64K Target RAM Module (part no. RDC-1020, RDC-1022 or RDC-1024, respectively) must also be installed for the PROM Programmer Module to operate. Refer to the Target RAM Module Installation Manual for details.

3.2 PROM PROGRAMMER INITIALIZATION

Module initialization and entry procedures for the LCE PROM Programmer Module are contained in this section. Both the softkey entry and PROM Programmer Module initialization procedures are described.

a. Turn-on LCE power (see Section 2.1 in the LCE System User's Manual).

b. Select the PRIMARY menu (see Section 2.2 in the LCE System User's Manual).

c. Press the PROM softkey number. The following PROM Programmer Module initialization message is displayed:

```
LCE PROM Programmer        Version 1.0

Rockwell International   copyright 1985
Supports 2K, 4K, 8K & 16K EPROMs
and 2K EEPROMs

PROM Number?
```

d. Enter the number of the device to be programmed and press RETURN. The system displays the selected PROM type and type of verification, e.g.:

```
2732A Selected
Verify While Prog.
```

An invalid device number entry causes the system to display the menu of supported PROMs and to redisplay the device type request prompt for a valid entry, e.g.:

```
Supported PROMs
2716 2532 2732 2732A 87C32 87C64 2564 2764 2764A 68764 68766
27128 27128A

Supported EEPROMs
2816 5213 48016

PROM Number?
```

The system will ask for verification of some PROM numbers, e.g., 2732A when 2732 is entered. Press Y or N to indicate proper device selection.

If a particular programming mode is selectable for the entered device type, the system will also prompt for entry of programming technique for some parts, e.g., for a 2764 or 2764A:

```
Pulse programmable?
```

Press Y or N to indicate the desired choice.

3.3 PROM PROGRAMMER MODES

After initialization or upon pressing ESC in one of the modes, the LCE displays these softkey major command modes heading:

1:MEMORY 2:PROM 3:UTILITY 4:SELECT 5:EXIT

The PROM Programmer modes are, in summary:

MEMORY - Performs memory dump, modification, inversion, load, save and verification functions.

PROM - Performs PROM check, program, read, compare and EEPROM erase functions.

UTILITY - Performs checksum, toggles verification mode and selects a new PROM type.

SELECT - Selects a new PROM type.

EXIT - Exits back to the LCE PRIMARY menu.

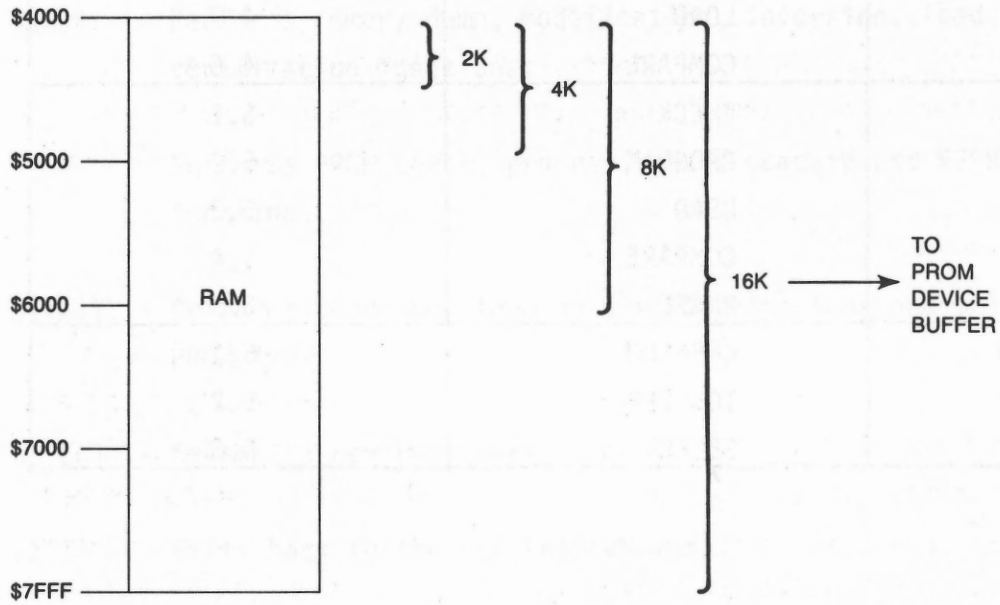
PROM commands perform operations involving a PROM installed in the PROM socket, e.g., READ PROM. PROM commands that transfer data from RAM to PROM (PROGRAM PROM), transfer data from PROM to RAM (READ PROM), or compare data between RAM and PROM (COMPARE PROM) use a block of RAM for data storage that is the size of the PROM. This part of RAM is called virtual PROM because it contains the contents of the PROM (Figure 3-1). This block of locations in RAM is specified upon command entry and must be contiguous within the RAM for the PROM type selected.

RAM commands affect RAM only and operate in any specified range of RAM. No check is performed to verify that the range of memory specified is actually in RAM.

Utility commands do not operate on RAM contents, but perform general purpose functions.

Table 3-1. PROM Programmer Commands

Category	Command	Section
MEMORY	DUMP	4.1
	MODIFY	4.2
	ALTER	4.3
	INVERT	4.4
	SAVE	4.5
	LOAD	4.7
	COMPARE	4.6
PROM	CHECK	5.1
	PROGRAM	5.2
	READ	5.3
	COMPARE	5.4
	ERASE	5.5
UTILITY	CHEKSUM	6.1
	TOG VER	6.2
	SELECT	6.3



AS REQUIRED, LOAD RECORDS WITH ADDRESSES OFFSET WHEN TRANSFERRING DATA INTO THE SELECTED PROM DATA BUFFER.

- FOR 2K DEVICES, THE PROM DEVICE DATA BUFFER ALWAYS BEGINS WITH A 2K-BYTE BOUNDARY ADDRESS (E.G., 1000, 1800, ETC.)
- FOR 4K DEVICES, THE ADDRESS STARTS ON A 4K-BYTE BOUNDARY ADDRESS (E.G., 1000, 2000, ETC.).
- FOR 8K DEVICES, THE ADDRESS STARTS ON AN 8K-BYTE BOUNDARY ADDRESS (E.G., 2000, 4000, ETC.).
- FOR 16K DEVICES, THE ADDRESS STARTS ON A 16K-BYTE BOUNDARY ADDRESS (E.G., 4000)

Figure 3-1. PROM Concept Example

3.4 GENERAL SYSTEM CHARACTERISTICS

There are several general characteristics of the LCE which pertain to all of the modes.

All keys are repeating keys. After holding a key down for approximately one second, the key continues entering characters until the key is released. This may occur even during the execution of a function, leading to a limited "type ahead" capability. The keystroke stack may be cleared by pressing "CTRL-SPACE BAR."

Whenever the system is performing a function, a cursor may not be displayed. Whenever the system is awaiting input, a flashing cursor is displayed.

3.4.1 Softkey Selection

The softkey concept is a software "relabeling" of the numeric keys. At any time when the system is ready for an operator to select a function, it displays a caption for up to eight numeric keys. Most of the modes do not require all eight options. In that case, the captions for those unassigned keys are left blank. The left most caption relates to the left most numeric key and so on for each of the keys. Functions called from a softkey menu usually return to that menu either on completion or an abort of the operation. The user choice is always indicated by the softkey label being displayed.

3.4.2 ESCAPE, MINUS and ZERO keys

When the system is in any function select mode, anticipating a softkey command, there are three additional keys which are significant. These are the ESCAPE key (ESC) and the MINUS (-) or ZERO (0) key. The ESCAPE key causes the next higher PROM Programmer command menu to be displayed. The MINUS or ZERO keys cause the set of options leading to the current set to be selected. They are, in effect, a "backspace one mode set" command. Any other key besides these keys and the captioned softkeys are ignored at function select time.

3.4.3 Address Entry

All target addresses and other data entered in response to prompts are assumed to be in hexadecimal, unless specified elsewhere by the input prompts.

The range to be tested must be within the limits of the RAM. If the range of addresses entered is either greater than applicable for the RAM, the system displays:

Range Too Large

If the "To" value is less than the "From" value, the system displays:

Negative Range

In either case, enter the correct "From" and/or "To" address and continue.

3.4.4 Command Verification

The prompt:

Are you sure?

expects either an upper or lower case "Y" as the affirmative response. Any other entry is considered a negative response.

3.4.5 In Process Indication

When any action takes place which requires longer than 15 seconds to accomplish, the system indicates that it is functioning by displaying a new decimal point or updating the address at least every 15 seconds. This assures you that the computer is functioning properly and may also be used to estimate the time to completion.

3.4.6 Improper PROM Insertion

If the PROM is installed in the wrong socket pins, the following messages may be displayed during PROM checking or programming functions:

```
Improperly inserted
Press any Key to Go
```

If this occurs, remove the PROM and insert it in the correct socket pins then press any key to continue.

SECTION 4

PROM PROGRAMMER MEMORY COMMANDS

This section describes the options available after the MEMORY mode has been selected. The captions over the softkeys are reassigned to the following names:

1:DUMP 2:MODIFY 3:ALTER 4:INVERT 5:SAVE 6:LOAD 7:VERIFY

The MEMORY functions are, in summary:

DUMP - Dumps a specified number of bytes in hex format starting at an entered address to a specified output device.

MODIFY - Modifies contents of memory starting at a specified address.

ALTER - Changes all locations between two addresses to a selected value.

INVERT - Logically inverts all locations between two addresses.

SAVE - Stores an image between two addresses, with an selectable offset, to a selected output device.

LOAD - Loads an object file, with selectable offset, from a specified input device.

VERIFY - Compares byte for byte between a file and memory.

4.1 DUMP

The DUMP function displays data or transfers data in target or prototype memory to an I/O device attached to either port, in a format suitable for viewing (ASCII and hexadecimal).

When the DUMP option is selected, the system displays the following:

```
From=$[S]hhhh
```

where: option S selects Single Byte
hhhh is a 4-byte hexadecimal address or a valid label (preceded by a ";"), followed by a RETURN.

4.1.1 Standard Dump

If only a hexadecimal address or label is entered, then the system prompts the user for the number of 16-byte lines to dump:

```
Count=dd
```

where: dd is a decimal value; 00 is continuous. A label preceded by a ";" can also be used.

Following this entry the system prompts the operator for the output device:

```
Output to (1/2) ?_   [1, 2, RETURN, or SPACE]
```

Once the output device has been entered, the system outputs the data in the following format:

```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
F400    41 20 73 61 6D 70 6C 65 20 6C 69 6E 65 0D F5 BE  A sample line...
F410    77 68 69 63 68 20 63 61 6E 20 77 72 61 70 20 20  which can wrap
F420    61 72 6F 75 6E 64 20 21 20 3C 3C 3C 3C 3C 3C 3C  around !
```

If the output port is the system console (RETURN or SPACE), pressing the SPACE bar causes the display to pause; pressing any key causes a paused display to continue. The output display speed can be changed using the 0-9 keys; 9 being the slowest and 0 being the fastest.

After the completion of the dump, or the pressing of the ESC key, the system returns to the MEMORY menu.

4.1.2 Single Byte Option

The single byte option is selected by pressing the S key in response to the "From" prompt. The address is then reprompted:

```
Single Byte   From=$hhhh
```

where: hhhh is a 4-byte hexadecimal number or a valid label, followed by a RETURN.

The prompt asks for the address to be displayed continuously after which the output port is requested.

```
Output to (1/2) ?_  [1, 2, RETURN or SPACE]
```

Once the output port is entered, the system displays the requested location, e.g.:

```
0018  00  .
0018  85  .
0018  07  .
0018  8C  .
0018  11  .
0018  93  (ESC)
```

Pressing the ESC key terminates the function.

4.2 MODIFY

The MODIFY function allows data in target or prototype memory to be changed, in either hexadecimal or ASCII format.

The system asks for the address of the first byte in memory to modify:

```
From=$[S,N] hhhh
```

where: option S selects Single Byte option,
option N selects No Read option,
hhhh is a 4-byte hexadecimal address or a valid label, followed by a RETURN.

STANDARD MODIFY

If just an address or label is selected, the system displays 16 bytes of data, in hexadecimal format, starting at that address. The data is also displayed in ASCII format to the right of the hexadecimal data in the manner as the DUMP command. The cursor is positioned to the left of the byte to be changed. The user can then:

- Move to the next byte by pressing the SPACE bar. This leaves the skipped data byte in memory unchanged.
- Modify the contents of the current data byte by typing a two-character hexadecimal value.
- Enter ASCII coded data by first pressing the T key. Note that the hexadecimal value corresponding to the entered ASCII data will not be displayed until the address is reselected.
- Backup one byte entry, or ASCII character, by pressing the left arrow or DEL key.
- EXIT the MODIFY function by pressing RETURN or ESC.

4.3 ALTER

Often, there is data in unused or uninitialized memory locations included in the block of data to be programmed into the PROM. For example, when a program is assembled or compiled such that the generated object code is output to a peripheral device, object code is usually not generated between the end of the program and the upper PROM boundary. If the object code file is subsequently loaded into RAM without first filling RAM with a known bit pattern and a PROM is programmed throughout the entire PROM range of addresses, random data is programmed into the unused locations. If the data file is subsequently loaded into RAM (at a different location or the RAM had been altered since the initial load) an invalid verification error may be indicated because new random data in unused locations in RAM are being compared with old random data previously programmed into the PROM.

To prevent invalid verification errors, it is advantageous to fill unused memory locations in RAM with a known bit pattern prior to loading the data to be programmed into RAM. Locations not loaded then retain the filled value.

A $00 bit pattern is often used because it is the 6500 CPU BRK operation code. However, $FF is also commonly used for a filler code because bits are left unprogrammed (the initial state of the erased PROM). By using the ALTER command, you can fill these unused RAM locations with a selected bit pattern.

a. After pressing the ALTER softkey, the system prompt is:

    Set Memory To = $00

b. Enter the two-character filler code, e.g., $FF, then press RETURN. The system then responds with:

    From=$hhhh (first address to be altered)

c. Enter the initial RAM address where the filler pattern is to be loaded, then press RETURN. The system response is:

```
To=$hhhh          (last address to be altered)
```

d. Enter the last RAM address where the filler pattern is to be loaded, then press RETURN (no range test is made). The system displays base address (modulo 256) of the RAM being changed. After the pattern is entered in the RAM throughout location specified, the system responds with:

```
Done
```

Example: Load $FF into RAM locations $E000 to $E4FF.

```
 Set Memory To = $FE
From=$E000 To=$E400
E400
 Done
```

If a memory error is detected, the following typical error message is displayed:

```
Memory error at $1000=20, should be $00
TARGET RAM ERROR
```

followed by display of the MEMORY menu.

4.4 INVERT

The INVERT function inverts (one's complement) data values in memory.

a. Press the INVERT softkey number. The system response is:

```
 Invert Memory
From=$hhhh        (first address to be inverted)
```

b. Enter the first RAM address where data is to be inverted, then press RETURN. The next prompt is:

```
To=$hhhh        (last address to be inverted)
```

c. Enter the final address of the range of RAM to be inverted, then press RETURN. At the completion of the inversion, the response is:

```
Done
```

Example: Inverts bits from $E000 to $E500:

```
 Invert Memory
From=$E000 To=$E500
E500
 Done
```

4.5 SAVE

The SAVE function creates an object code file of data in target/prototype memory and saves it to a user-selected output device. The file is sent in either semicolon or colon format, depending on the SYSGEN parameter selected on the port in use.

a. Press the SAVE softkey number. Upon selection, the system subsequentially asks for the address range of target/prototype memory to save, followed by the offset.

```
From=$hhhh   To=$hhhh  Offset=$hhhh
```

where: hhhh is a 4-byte hexadecimal value or a valid label, followed by a RETURN.

b. Following the entry of these parameters, the system asks if there are any other areas of target/prototype memory to be saved:

```
More  (Y/N)?_ [Y or N]
```

If Y is pressed, additional prompts for addresses and offset are displayed. If N is pressed, the system then asks for the output port:

```
Output to (1/2) ?_ [1, 2, RETURN or SPACE]
```

c. Once the output device is accepted, the LCE issues a start character (if one is assigned) and begins to issue object records. A "." is displayed on the console for each record issued, unless the output device is the system console. After all records are transferred, a stop character is issued (if one is assigned) and the system returns to the MEMORY menu.

4.6 LOAD

The LOAD function loads target/prototype memory with object code from either I/O port. The LOAD option accepts semi-colon, colon, or block colon formats.

a. Upon selection, the system first prompts:

```
Offset=$hhhh
```

where: hhhh is a 4-byte hexadecimal value or a valid label, followed by a RETURN.

The offset value is a displacement to the address of the object code file into target/prototype memory.

b. The system then asks for the input device:

```
Input from (1/2) ? [1, 2, RETURN or SPACE]
```

NOTE

RETURN or SPACE defaults to Port 1.

c. Once the input device is accepted, the LCE issues a start character (if one is assigned) and awaits the object data. A "." is displayed on the console for each record received correctly. A "*" is displayed for each record in error (typically a checksum error or invalid memory). After a transfer of all records, or receipt of a stop character (if one is assigned), the system returns to the MEMORY menu.

NOTE

If the object records are sent too fast for the LCE, then a "Port Overrun Error" message will be displayed and the Load function will be aborted. If this occurs, try using either a slower baud rate, or adding delays between object records.

4.7 VERIFY

The VERIFY function compares a selected object code file with data in target/prototype memory. The VERIFY option accepts semi-colon, colon, or blocked colon formats.

a. Upon selection, the system first prompts:

Offset=$hhhh

where: hhhh is a 4-byte hexadecimal value or a valid label, followed by a RETURN.

The offset value is a displacement added to the address at the object code received for verification in target/prototype memory.

b. The system then asks for the input device:

Input from (1/2) ?_ [1, 2, RETURN or SPACE]

c. Once the input port is accepted, the LCE issues a start character (if one is assigned) and awaits the object data. A "." is displayed on the console for each record received correctly. A "*" is displayed for each record in error (typically a checksum or byte mismatch). After verification of all records, or receipt of a stop character (if one is assigned), the system returns to the MEMORY menu.

SECTION 5

PROM PROGRAMMER PROM COMMANDS

This section describes the available options after the PROM mode has been selected. The captions over the softkey numbers are reassigned to the following names:

1:CHECK 2:PROGRAM 3:READ 4:COMPARE 5:ERASE

5.1 CHECK

When the PROM is initialized, all PROM locations contain "1's." The CHECK command tests whether or not the PROM contains all "1's" in a specified range of memory. To use the CHECK command:

a. Press the CHECK softkey number. For a 2532 device, the system displays:

```
2532 Selected
From=$hhhh              (first address to be checked)
```

b. Enter the beginning PROM address and press RETURN. The next prompt is:

```
To=$hhhh                (last address to be checked)
```

c. Enter the last PROM address of the range of PROM locations to be checked, then press RETURN. If the PROM is not initialized, the response is:

```
2532 Selected
From=$XXXX To=$YYYY
NOT INITIALIZED
```

If the PROM is initialized, the system response is:

```
2532 Selected
From=$XXXX To=$YYYY
 Done
```

Example 1: 2532 PROM initialized, $1000 and $1FFF for "From" and "To" address.

```
2532 Selected
From=$1000 To=$1FFF
1F00
 Done
```

Example 2: 2532 PROM not initialized, $1000 and $1FFF for "From" and "To" addresses, respectively.

```
2532 Selected
From=$1000   To=$1FFF
1F00
NOT INITIALIZED
```

5.2 PROGRAM

The PROGRAM command copies RAM contents (in the range of memory locations specified) into the PROM. First, the PROM is checked to determine if the PROM is initialized within the range specified. If the PROM is not initialized, an option is offered to either continue or exit from the programming routine. If the choice is to continue, the PROM is programmed. Contents of the PROM are verified during programming (Verify While Prog.) or after programming (Verify After Prog.) depending on which mode is selected (via the toggle VERIFY command). This VERIFY mode selection must be done prior to selecting the programming function and the PROM capabilities. The mode of verification is displayed when the programming function is selected.

If "Verify While Prog". is selected, contents of each byte in RAM are compared to the corresponding contents of the PROM throughout the programming range both prior to and after programming the byte. If the data is the same prior to programming, programming of that byte into PROM is omitted and the processing is advanced to check the next byte. Therefore, programming is performed at a faster rate when programming of all bytes is not required. As a result, the ALTER command should first be used to fill RAM with all "1's" throughout the programming range. This ensures that after loading the data to be programmed into RAM, all initialized RAM bits correspond to initialized PROM bits.

During the "Verify While Prog." mode, a check is also made to determine if each bit in an EPROM (not EEPROM) can be programmed, i.e., that an attempt is not made to write a "1" bit value in RAM into a "0" bit value in PROM. If such an attempt is made, the message:

```
Invalid Bit Change At XXXX    (XXXX = hex address)
```

is displayed, the programming function terminates, the "Done" message is displayed, and control returns the the PROM command menu level.

Also, in the "Verify While Prog." mode, each byte is checked after programming to verify proper PROM coding. If the value in the PROM is not identical to the value in the RAM, the message:

```
ADDRESS PROM RAM
*XXXX    YY  ZZ
```

is displayed to show the address of the error with the corresponding PROM and RAM values.

When "Verify After Prog." is selected, contents of the PROM are compared to contents of RAM after programming of all bytes in the specified range is complete. If any differences are detected, the error message heading:

```
ADDRESS PROM RAM
```

is displayed followed by the address of each nonmatching value and the values in the form:

```
*XXXX    YY  ZZ
```

In some instances the PROM cannot be verified while programming, e.g., the 68764. When such a PROM is being programmed, the "Verify After Prog." mode is automatically selected.

NOTE

It is possible to change a bit value from "1" to "0" in the EPROM by using the PROGRAM function, however, the EPROM must be exposed to an ultraviolet light in accordance with the manufacturer's instructions to reset the bit value back to "1" (i.e., erased). If the EPROM is not initialized, programming is performed correctly only if the new code does not contain any "1's" in the bit positions already set to "0's".

a. To use the PROGRAM command, press the PROGRAM softkey number. The system responds with:

```
2816 Selected
Verify While Prog.  (or Verify After Prog.)
From=$hhhh          (first address to be programmed)
```

where: 2816 is the PROM type number used in this example.

NOTE

The full address range of the PROM must lie on a RAM range of that modulo, i.e., a 4K PROM boundary to be on a 4K RAM boundary.

b. Enter the first RAM address to be programmed then press RETURN. The next prompt is:

```
To=$hhhh          (last address to be programmed)
```

c. Enter the last RAM address to be programmed and press RETURN.

d. The system then asks where error messages are to be displayed:

```
Errors Output to (1/2) ?
```

e. Press either 1 or 2 as applicable to specify whether the error message is to be sent to Port 1 or Port 2.

f. The PROM is first checked to determine if it is initialized. If all bits throughout the selected programming range are "1's," programming continues (see step g). If any of the bits are "0's," the message:

```
NOT INITIALIZED Continue?
```

is displayed prior to continuing. If this message is displayed, press "Y" (for yes) to continue the programming function (see step g) or "N" (for no) to terminate the programming function and return control to the PROM command menu level.

g. The PROM is now programmed in accordance with the PROM type specification. During the programming process, a four-digit hexadecimal number is displayed. This number represents the part of memory being programmed and changes about every second. The number displayed depends upon the PROM type and address range programmed. The time required to completely program each PROM throughout the complete device address range is as follows (approximately):

PROM Type	Programming Time
2532	200 seconds
2564	200 seconds
2716	100 seconds
2732	200 seconds
2732A	186 seconds
87C32	200 seconds
2764	180 seconds
2816A	60 seconds
5213	60 seconds
27128	360 seconds
48016	30 seconds
68764	130 seconds

h. The message "Done" is displayed upon completion of the programming.

NOTE

Some PROMs cannot be verified during programming because of timing constraints (check the PROM specifications). If repeated verification errors are indicated, escape from the programming function to the top level PROM Programmer menu, select the UTILITY menu, toggle the VERIFY mode with the TOG VER function to verify after programming, then reenter the program PROM function.

Example 1: Program a 2532 PROM from $1000 to $16FF (no errors) and verify while program.

```
2532 Selected
Verify While Prog.
From=$1000 To=$16FF
16F0
Errors Output to (1/2) ?
 Done
```

Example 2: Program a 2532 PROM from $1500 to $15FF (errors at $1530 to $1535) and verify after programming.

```
2532 Selected
Verify After Prog.
From=$1500 To=$15FF
Errors Output to (1/2) ?
Verifying

ADDRESS PROM RAM

*1530    FF  00
*1531    FF  00
*1532    FF  00
*1533    FF  00
*1534    FF  00
*1535    FF  00

 Done
```

5.3 READ

The READ command reads PROM contents into RAM in the range of RAM addresses specified.

a. Press the READ softkey number to enter the READ mode. The system responds with:

```
2532 Selected    (2532 used in this example)
From=$hhhh       (first address to be read)
```

b. Enter the first address in RAM where contents are to be stored when read from the first PROM address, then press RETURN. The next prompt is:

```
To=$hhhh         (last address to be read)
```

c. Enter the last address in RAM where contents of the last PROM location are to be stored, then press RETURN. (The same address range test is made for the READ command as for the PROGRAM command). The system then reads PROM contents into the RAM addresses specified. When the READ process is complete, the read completed message is displayed:

```
Done
```

d. If the contents of RAM do not equal the contents of the PROM after the read is completed, the following error message is displayed:

```
Memory error at $XXXX=$HH, should be $JJ
  TARGET RAM ERROR
```

where: XXXX = hex address
HH = RAM value
JJ = PROM value

Example: Read a 2532 PROM into $1000-$1FFF.

```
2532 Selected
From=$1000 To=$1FFF
 Done
```

5.4 COMPARE

The COMPARE function compares PROM contents with RAM contents to verify either that RAM data has been programmed correctly into the PROM, or that PROM data has been correctly read into RAM. (This verification function is also performed automatically either during or after programming a PROM.) If data is not identical in corresponding PROM and RAM locations, RAM addresses are shown with corresponding PROM and RAM data bytes.

a. Press the COMPARE softkey number to enter the COMPARE function. The system prompts:

```
2532 Selected      (2532 used in this example)
From=$hhhh         (first address to be compared)
```

b. Enter the first RAM address where data from the first PROM address is to be compared, then press RETURN. The next prompt is:

```
To=$hhhh           (last address to be compared)
```

c. Enter the last RAM address of the range of locations where data is to be compared, and press RETURN. (The same address range test is made for the COMPARE command as for the PROGRAM command.) The next prompt is:

```
Errors Output to (1/2) ?
```

d. Press either 1 or 2 to indicate which port the errors are to be output.

The contents of RAM are compared to the contents of the PROM. When PROM and RAM data differences occur, values and addresses are displayed as shown in the following example:

```
ADDRESS PROM RAM

*3000    00  07
*3001    00  EF
*3002    00  04
```

After completion of the COMPARE function, the system displays:

```
 Done
```

Example 1: Compare a 2532 PROM from $3000 to $3FFF (no errors).

```
2532 Selected
From=$3000 To=$3FFF
Errors Output to (1/2) ?
3FFF

 Done
```

Example 2: Compare a 2532 PROM from $E000 to $EFFF (with errors).

```
2532 Selected
From=$E000 To=$EFFF
Errors Output to (1/2) ?

ADDRESS PROM RAM
*E000    00  07
*E001    00  EF
*E002    00  04

EF00

 Done
```

5.5 ERASE

EEPROMs are electrically erasable PROMs with nonvolatile memory.

a. After pressing the ERASE softkey number, the system displays:

    2816 Selected Are You Sure? (2816 is used in this example)

    If the PROM selected is not an EEPROM, the system typically displays:

    2716 Selected, Use UV Lamp!

    In this example, the 2716 PROM is not electrically erasable and can only be erased by ultraviolet light.

b. Press "N" to escape the function or, "Y" to erase the EEPROM contents. If "N" is pressed, the erase function terminates and control returns to the PROM menu command level. If "Y" is pressed, the erase function continues and the response is:

    From=$hhhh (first address to be erased)

b. Enter the PROM address where erasure is to start and press RETURN. The next prompt is:

    To=$hhhh (last address to be erased)

c. Enter the last address of the range of memory locations to be erased and press RETURN. The EEPROM is erased over the selected address range. On completion, the system response is:

    Done

NOTE

While erasing is being completed, a four-digit hexadecimal value is displayed, and increased in $100 increments.

If an EEPROM cannot be completely erased after using the ERASE function once, repeat the ERASE function as required.

Example:

```
2816 Selected  Are You Sure?Y  (2816 used in this example)
From=$E100 To=$F2FF
E2F0
E200
 Done
```

5.6 COPYING A PROM

PROM contents can be copied into another initialized PROM of the same type by using the following procedure (with control at the PROM Programmer Module command level):

a. Insert the PROM to be copied from into the PROM ZIF socket (Figure 2-3).

b. Use the READ command to read the PROM data into the RAM in the address range specified.

c. Use the SELECT command to initialize the PROM Programmer for the applicable PROM type to be programmed.

d. Remove the PROM from the PROM ZIF socket and insert the initialized PROM (to be copied to) into the PROM ZIF socket.

e. Use the PROGRAM command to program the same range of RAM contents into the PROM.

f. Remove the copied PROM.

SECTION 6

PROM PROGRAMMER UTILITY COMMANDS

This section describes the available options after the UTILITY menu has been selected. The captions over the softkeys numbers are reassigned to the following names:

1:CHEKSUM 2:TOG VER 3:SELECT

The UTILITY commands are, in summary:

CHEKSUM - Computes the checksum of PROM contents over a specified address range.

TOG VER - Toggles the PROM programming verify mode.

SELECT - Selects the PROM part type.

6.1 CHEKSUM

During the execution of the CHEKSUM command, data in RAM throughout a specified address range is added iteratively in 8-bit segments (bytes). The resulting sum (the checksum) is displayed as a 16-bit value. The CHEKSUM command can be used to determine if PROM contents have been loaded correctly by recording the checksum value generated for a selected range of RAM data to be written into the PROM. Subsequently, use the READ command to read the PROM data back into RAM at a new range of addresses, and then use the CHEKSUM command to generate a new checksum value for data at that range of addresses. If the two checksum values match, the PROM has been programmed correctly.

a. To use the CHEKSUM command, press the CHEKSUM softkey number. The system responds with:

```
From=$hhhh          (first address to be summed)
```

b. Enter the beginning RAM address and press RETURN. The next prompt is:

```
To=$hhhh            (last address to be summed)
```

c. Enter the ending RAM address and then press RETURN. The system response is:

```
XXXX                (during checksum calculation only)
Checksum = NNNN
 Done
```

where NNNN is the 16-bit checksum hex value.

Example:

```
From=$E000 To=$E002
EXXX                (during checksum calculation only)
Checksum = 02FD
 Done
```

The 02FD is the checksum value obtained when an all 1's pattern is loaded in RAM locations $E000, $E001, and $E002.

6.2 TOG VER

The TOG VER command causes verification to occur either while programming or after programming.

a. After pressing the TOG VER softkey, the system responds with:

Verify While Prog.

or

Verify After Prog.

depending on the previously selected mode.

Example: The following is displayed after pressing TOG VER two times (verify after Prog. was initially selected).

Verify While Prog.

Verify After Prog.

6.3 SELECT

The SELECT command either initially enters or changes a previously entered PROM type number.

a. After pressing the SELECT softkey number, the system response is:

PROM Number?

b. Enter the desired PROM part number and press RETURN.

c. If an illegal PROM type number, e.g., 2888, is entered the system displays:

```
PROM Number? 2888
**ERROR**

Supported EPROMs
2716 2532 2732 2732A 87C32 87C64 2564 2764 68764 68766 27128 27128A

Supported EEPROMs
2816 5213 48016

PROM Number?
```

In this case, re-enter the correct part number.

SECTION 7

FUNCTIONAL AND INTERFACE DESCRIPTIONS

7.1 MEMORY MAP

The LCE PROM Programmer software is contained in an 8K PROM (ID J907-001) and is installed in socket Z16 on the LCE PROM Programmer Module. Addresses $A000-$BEFF are assigned to program instrurctions and $BF00-$BFFF are dedicated to module I/O. The addresses used for I/O are $BF00 (DAC), $BF10-$BF1F (VIA), $BF20 (Latch 1) and $BF30 (Latch 2). Other addresses allocated to I/O are not used.

A memory map of the LCE including the PROM Programmer Module is illustrated in Figure 7-1. PROM Programmer Module addresses are listed in Table 7-1.

Table 7-1. PROM Programmer Module Memory Map

Device	Read/Write	Addresses
ROM	Read Only	X000 to XFFF excluding XY00 to XYFF
DAC	Write Only	XY00 to X0FF, XY40 to XY4F, XY80 to XY8F, XYC0 to XYCF
VIA	Read/Write	XY10 to XY1F, XY50 to XY5F, XY90 to XY9F, XYD0 to XYDF
LATCH 1	Write Only	XY20 to XY2F, XY60 to XY6F, XYA0 to XYAF, XYE0 to XYEF
LATCH 2	Write Only	XY30 to XY3F, XY70 to XY7F, XYB0 to XYBF, XYF0 to XYFF

X = Status of S1-5 to S1-8 = A for LCE PROM Programmer ROM
Y = Status of S1-1 to S1-4 = F for LCE PROM Programmer ROM

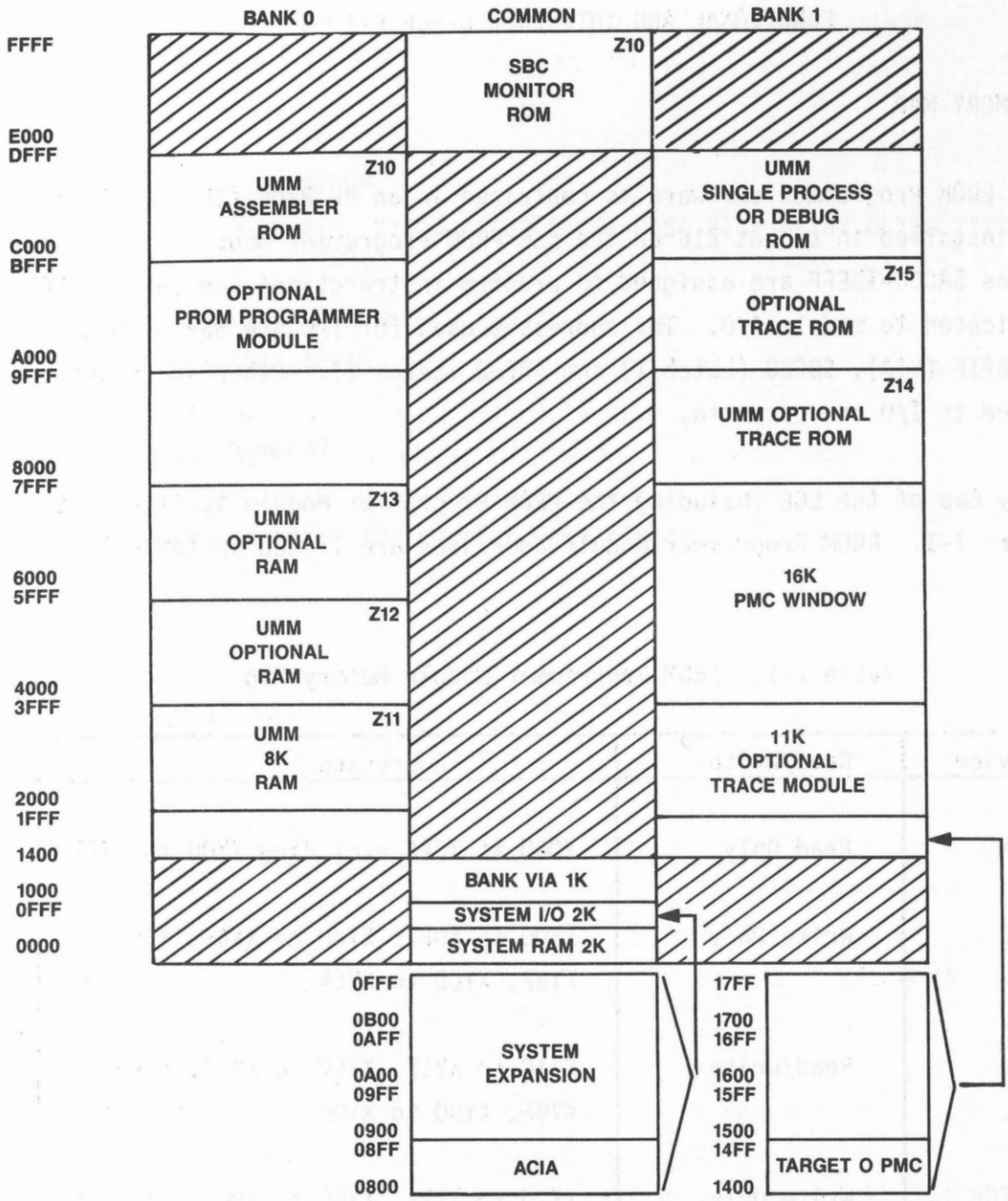


Figure 7-1. LCE Memory Banking Map

7.2 FUNCTIONAL DESCRIPTION

This functional description is based on the block diagram of the PROM Programmer Module (Figure 7-2), and the PA10-X441 schematic at the end of this document.

Data Bus Transceivers

Data Bus Transceivers isolate the PROM Programmer Module internal data bus from the RM 65 Bus. The Data Bus Transceivers gate RM 65 Bus data into the module during a write operation and gate module data onto the RM 65 Bus during a read operation. Also, Data Bus Transceiver circuits invert data to allow the module to operate internally with noninverted data.

Address and Control Signal Buffers

All address and control signals are buffered and inverted for internal use on the PROM Programmer Module.

Address Signal Buffers

ROMADD is derived from Chip Select Decode circuitry and jumper E1. When E1 is in the B position, ROMADD is disabled (no ROM on board).

PHASE2, R/W, A4, and A5 inputs to the Program Array Logic (PAL) correspond directly to those signals from the control and address signal buffers. R/W is incorporated to ensure that write-only devices are not enabled during a write operation. PHASE2 gates all device enables to allow ample device deselect time.

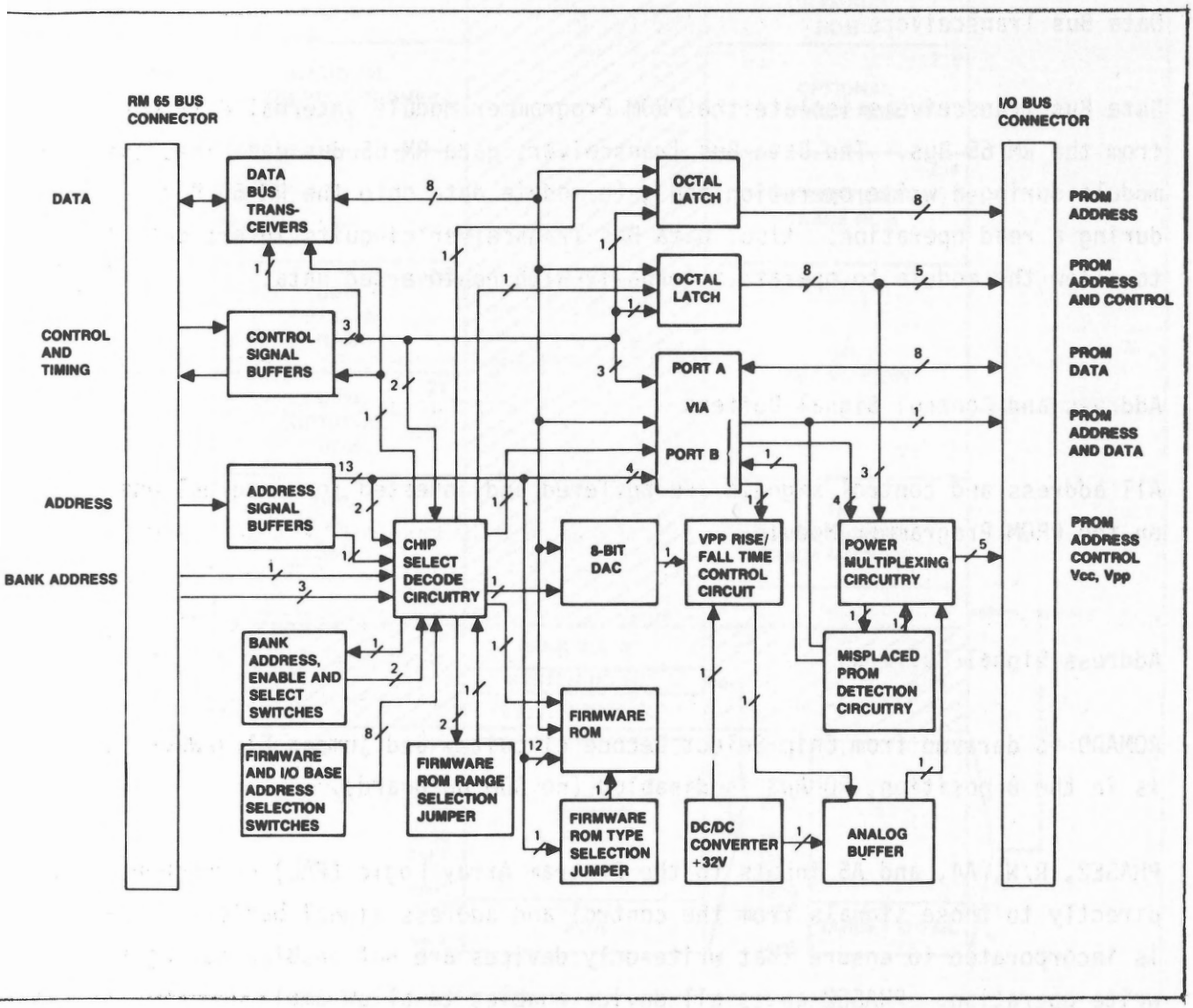


Figure 7-2. PROM Programmer Module Block Diagram

Chip Selection Decode Circuitry

Chip Select Decode circuitry includes a PAL which is programmed to enable the device specified by the address signals. When signals of the most significant byte of the address match signals from Bank Address, Enable, and Select Switches S1-1 through S1-8, the I/O page is addressed.

Bank Address Enable and Selection Switches

Switches S1-9 and S1-10 select the memory bank(s) to be used with the module. If switch S1-10 is open, the state of BADR/ is ignored and data is entered into either bank (see Table 2-3).

Firmware, I/O Base Address Selection Switches and Jumpers

On-board programs are stored in the LCE PROM number J907-001. Before using the PROM Programmer Module, E1 and E2 jumpers must be set to enable the on-board ROM (refer to 2.2.3).

Switches S1-1 to S1-8 control the firmware and I/O base addresses of memory locations used with the PROM Programmer Module. Switches S1-5 to S1-8 are set to the base address of the on-board ROM. Switches S1-1 to S1-4 are set to indicate the one-page I/O base address of the module registers within the ROM address space. For each base address, a switch in the on (down) position represents a logic one. Switches S1-8 and S1-4, respectively, represent the firmware and I/O base address most significant bits.

Jumper E1 controls addressing of the on-board ROM. When E1 is in the B position, no ROM space is selected and the firmware resides off board in the LCE.

Programmable Voltage Regulator

The Programmable Voltage Regulator generates the programming voltage. The regulator consists of the 8-bit DAC, the VPP Rise/Fall Time Control circuit, the DC-to-DC Converter, and the Analog Buffer.

The DAC controls the amplitude of the VPP programming voltage. The DAC output voltage ranges from 0 to +2.55 Vdc which is amplified from 0 to 25.5 Vdc. The programming voltage is generated by writing an 8-bit word into the DAC. The decimal value of the word corresponds directly to the voltage generated, e.g., 250=$FA=25 Vdc.

For certain EPROM's, the programming voltage must rise and fall at a specified rate. Discrete analog components connected to the DAC amplifier ensure that the VPP rise and fall times are between 0.5 and 2.0 microseconds. These circuits also prevent the programming voltage from going lower than 5 Vdc. This protection is required because some EPROM's must have the VPP pin held at 5 Vdc when they are read. After the programming voltage is generated by the DAC, a controlled rise or fall time can be initiated by writing either a zero or a one, respectively, into bit 5 of the VIA B port.

The Analog Buffer serves as a current amplifier for the VPP output of the rise/fall circuitry. The output of the Analog Buffer drives the VPP circuits in the power multiplexing circuitry.

The DC-to-DC Converter is a power supply that steps up 5 Vdc to 32 Vdc. This voltage is used by the Programmable Voltage Regulator to generate the programming voltage.

Versatile Interface Adapter (VIA) and Octal Latches

The R6522 Versatile Interface Adapter (VIA) reads data from and writes data to the PROM being programmed and controls the programming voltages level. Port A of the VIA is connected to the data lines of the programmed device. Pins PA0 through PA7 are connected to the Power Multiplexing Circuit, the VPP Rise/Fall Time Control Circuit, and the device program control line. Pins PB0 through BP3 are connected to the Power Multiplexing Circuit. Pin PB4 is not connected. Pin PB5 is connected to the VPP Rise/Fall Time Control Circuit. Pin PB6 is connected to the program line PGM/. Pin PB7 is connected to the Misplaced PROM Detection Circuit.

The Octal Latches generate the addressing for the programmed device and latch the level of the programming voltage. Latch A, Pins LA0, LA1, LA2, LA4, and LA6 of Latch A are connected to address lines A8, A9, A10, A12, and CE/, respectively. Pins LA3, LA5, and LA7 of Latch A are connected to the Power Multiplexing Circuitry. Pins LB0 through LB7 of Latch B are connected to address lines A0 through A7, respectively.

Power Multiplexing Circuitry

The Power Multiplexing circuitry selects the correct voltage levels to be output on pins 1, 2, 29, 31, 36, and 40 of connector P2. The voltages that are selected are TTL HIGH, TTL LOW, VCC and the output voltage of the Analog Buffer. The selected voltage is specified by the states of control signals from the VIA and Octal Latch A. Output voltages on the P2 connector pins for a given set of control signal states are listed in Table 7-2.

Misplaced PROM Detection Circuitry

The Misplaced PROM Detection Circuitry determines if a 24-pin device has been misplaced in the 28-pin Zero Insertion Force socket. If a 24-pin device is incorrectly inserted (offset by one or two pins), then the VCC pin of the device is detected by the current flowing out of P2 pins 38 or 40. This condition causes a HIGH state which can be read on the VIA port line PB7. If the 24-pin device is correctly installed, a LOW state is read on this line.

CAUTION

The Misplaced PROM Detection Circuitry is effective only for devices that are incorrectly placed into the ZIF socket with a one- or two-pin offset. The circuit can not detect when a device is installed upside down.

Table 7-2. Output Voltage Selections

P2 Output Pin	Condition	Output
2	PB0-Low PB0-High	Vpp High Impedance
29	PB2-Low, LA7-High PB2-High, LA7-High PB2-X, LA7-Low	Vpp TTL High TTL Low
31	PB1-Low, LA3-High PB1-High, LA3-High PB1-X, LA3-Low	Vpp TTL High TTL Low
36	LA5-Low LA5-High	TTL Low Vcc
1,40	PB3-Low PB3-High	Vcc High Impedance

7.3 INTERFACE DESCRIPTION

Two interfaces exist on the PROM Programmer Module, i.e., the P1 connector to the RM 65 Bus and the J1 connector that interfaces with the Socket Module cable and Socket Module connector. P1 connector-pin signal assignments are listed in Table 7-5; signal descriptions for J1 pins and PROM Programmer Socket module socket-pin numbers are furnished on schematic PA10-X441 (at the end of this document). Similar information is listed in Table 7-4.

Table 7-3. Connector P1 (RM 65 Bus) Pin Assignments

Pin	Signal Mnemonic	Signal Name	Input/Output
1a	GND	Ground	
1c	+5V	+5 Vdc	
2a	BADR/	Buffered Bank Address	I
2c	BA15/	Buffered Address Bit 15	I
3a	GND	Ground	
3c	BA14/	Buffered Address Bit 14	I
4a	BA13/	Buffered Address Bit 13	I
4c	BA12/	Buffered Address Bit 12	I
5a	BA11/	Buffered Address Bit 11	I
5c	GND	Ground	
6a	BA10/	Buffered Address Bit 10	I
6c	BA9/	Buffered Address Bit 9	I
7a	BA8/	Buffered Address Bit 8	I
7c	BA7/	Buffered Address Bit 7	I
8a	GND	Ground	
8c	BA6/	Buffered Address Bit 6	I
9a	BA5/	Buffered Address Bit 5	I
9c	BA4/	Buffered Address Bit 4	I
10a	BA3/	Buffered Address Bit 3	I
10c	GND	Ground	
11a	BA2/	Buffered Address Bit 2	I
11c	BA1/	Buffered Address Bit 1	I
12a	BA0/	Buffered Address Bit 0	I
12c	B01	Not Used	
13a	GND	Ground	
13c	BSYNC	Not Used	
14a	BSO	Not Used	
14c	BDRQ1/	Not Used	
15a	BRDY	Not Used	
15c	GND	Ground	
16a		Not Used	
16c		Not Used	
17a	+12V	Not Used	
17c		Not Used	
18a	GND	Ground	
18c		Not Used	
19a	BFLT/	Not Used	
19c	B00	Not Used	
20a		Not Used	
20c	GND	Ground	
21a	BR/$\overline{W}$/	Buffered Read/Write "Not"	I
21c	BDRQ2/	Not Used	
22a		Not Used	
22c	BR/$\overline{W}$	Not Used	
23a	GND	Ground	
23c	BACT/	Buffered Bus Active	O
24a	BIRQ/	Not Used	
24c	BNMI/	Not Used	
25a	B02/	Buffered Phase 2 "Not" Clock	I
25c	GND	Ground	
26a	B02	Not Used	
26c	BRES/	Buffered Reset	I
27a	BD7/	Buffered Data Bit 7	I/O
27c	BD6/	Buffered Data Bit 6	I/O
28a	GND	Ground	
28c	BD5/	Buffered Data Bit 5	I/O
29a	BD4/	Buffered Data Bit 4	I/O

Table 7-3. Connector P1 (RM 65 Bus) Pin Assignments (Cont)

Pin	Signal Mnemonic	Signal Name	Input/Output
29c	BD3/	Buffered Data Bit 3	I/O
30a	BD2/	Buffered Data Bit 2	I/O
30c	GND	Ground	
31a	BD1/	Buffered Data Bit 1	I/O
31c	BD0/	Buffered Data Bit 0	I/O
32a	+5V	+5 VDC	
32c	GND	Ground	

NOTE
"/" suffix denotes signal active at negative or low voltage level.

Table 7-4. Connector P1 (RM 65 Bus) Signal Descriptions

Mnemonic	Signal Name and Signal Description
+5V	+5 VDC supplied to the module from the RM 65 Bus.
GND	**Ground** System ground (15c and 18a are excluded).
BA0/ – BA15	**Buffered Address Bits 0 – 15** BA0 through BA15 constitute an inverted 16-bit parallel address from the RM 65 Bus to the module.
BADR/	**Buffered Bank Address** A high BADR/ signal specifies the lower 65K (Bank 0) memory bank; a low BADR/ specifies the upper 65K (Bank 1) memory bank. Depending on S1 switch positions, BADR/ is either used as a seventeenth address bit or ignored by the address decode logic.
BACT/	**Buffered Bus Active** A low BACT/ indicates that the module is addressed and the Data Transceivers are enabled in either the receive (write operation) or transmit (read operation) direction.
BD0/ – BD7/	**Buffered Data Bits 0 – 7** BD0 through BD7 are the inverted data signals transferred as an 8-bit data byte between the Data Teceivers in the module and the RM 65 Bus.
B02/	**Buffered Phase 2 Clock "Not"** B02 synchronizes data transfers to and from the RM 65 Bus. The address and read/write signals are set up during the positive portion of B02/; data is transferred in the negative portion of B02/. B02/ is applied to the Chip Select Decode Circuitry and to the VIA.
BR/$\overline{W}$/	**Buffered Read/Write "Not"** BR/$\overline{W}$/ controls the direction of data transfers to and from the RM 65 Bus. A low BR/$\overline{W}$/ indicates a read operation. A high BR/$\overline{W}$/ indicates a write operation.
BRES/	**Buffered Reset** BRES/ is a hardware reset signal sent from the RM 65 Bus to the module. A low BRES/ clears the octal latches and initializes the VIA.

NOTE
All signals interfaced to and from the module are driven at TTL voltage levels.

Table 7-5. Connector J1 (PROM Interface) Pin Assignments

Connector J1 (P2) Pin No.	24-pin PROM Socket Pin Number (1)	28-pin PROM Socket Pin Number	Signal Symbol	Signal Name	Connector J1 (P2) Pin No.	24-pin PROM Socket Pin Number (1)	28-pin PROM Socket Pin Number	Signal Symbol	Signal Name
3,5,10,	12	14	GND	Ground	16	9	11	Q0	Data Bit 0
15, 24, 26					17	10	12	Q1	Data Bit 1
28, 30, 32					18	11	13	Q2	Data Bit 2
35, 37, 39					19	13	15	Q3	Data Bit 3
1, 40	—	28	VCC	PROM Supply Voltage	20	14	16	Q4	Data Bit 4
2	—	1	VPP	Programming Voltage	21	15	17	Q5	Data Bit 5
6	1	3	A7	Address Bit 7	23	17	19	Q7	Data Bit 7
7	2	4	A6	Address Bit 6	25	18	20	$\overline{CE}$	Chip Enable
8	3	5	A5	Address Bit 5	27	19	21	A10	Address Bit 10
9	4	6	A4	Address Bit 4	29	20	22	$\overline{OE}$	Output Enable
11	5	7	A3	Address Bit 3	31	21	23	A11	Address Bit 11
12	6	8	A2	Address Bit 2	33	22	24	A9	Address Bit 9
13	7	9	A1	Address Bit 1	34	23	25	A8	Address Bit 8
14	8	10	A0	Address Bit 0	36	24	26	A13/V_{CC}(2)	Address Bit 13
					38	—	27	$\overline{PGM}$	Program

Note: (1) Nomenclature applies to the recommended JEDEC Bytewide pin-out
(2) Applies to 24-pin devices

Table 7-6. Connector J1 (PROM Interface) Signal Descriptions

Mnemonic	Signal Name and Signal Description
GND	**Ground** Module ground.
Vcc	**PROM Supply Voltage** Vcc is the voltage for 28-pin PROM's. It is switchable to allow it to be disabled for safe insertion and removal of PROM's. VCC is disabled by a hardware reset.
Vpp	**Programming Voltage** Vpp is the high programming voltage for 28-pin PROM's. Vpp is programmable as described in paragraph 4.2.9. This output is switchable through firmware and disabled by a hardware reset.
A12	**Address Bit 12 (A12)** This signal is used either as address bit A12 or as a chip-select signal for 28-pin PROM's. A12 is set low by a hardware reset.
A0 to A9	**Address Bit 0 to 9** A0 to A9 are address signals common to all EPROM types. They are set low by a hardware reset.
Q0 to Q7	**Data Bit 0 – 7** Data bit signals Q0 to Q7 are bidirectional data signals common to all PROM types. These signals float high through a 2.8K ohm pull-up resistor when a hardware reset is asserted.
CE/	**Chip Enable** This signals is used as a chip-enable, programming pin, or address bit 11 signal as required by the PROM being programmed. It is set low by a hardware reset.

APPENDIX A
SCHEMATIC DIAGRAMS

This appendix contains the schematic diagrams for the PROM Programmer Module and PROM socket board used with the LCE system. These schematic diagrams are presented in this appendix as reference material only and are subject to change without notice if deemed necessary by Rockwell International to improve the product.

Module Name	Drawing No.	Page No.
PROM Programmer Module	PA10-X441	A-3
PROM Socket Board (installed on LCE Front Panel)	PA10-X444	A-5

REVISIONS			
LTR.	DESCRIPTION	DATE	APPROVAL
NC	TDR 530610	9-28-82	

SIGNAL FUNCTION / EPROM PIN NUMBER

J1

14	A0	/ 10
13	A1	/ 9
12	A2	/ 8
11	A3	/ 7
9	A4	/ 6
8	A5	/ 5
7	A6	/ 4
6	A7	/ 3
16	Q0	/ 11
17	Q1	/ 13
18	Q2	/ 14
19	Q3	/ 15
20	Q4	/ 16
21	Q5	/ 17
22	Q6	/ 18
23	Q7	/ 19
38	R/W	/ 27
34	A8	/ 25
33	A9	/ 24
27	A10	/ 21
4	A12	/ 2
25	CE	/ 20
2	VPP	/ 1
29	OE	/ 22
31	A11	/ 23
36	A13	/ 26
1	VCC	/ 28
40		
3		

SPARES:

ON PAL10L8 (Z11): 3 UNUSED INPUTS (8,9,11)
2 UNUSED OUTPUTS (18,19)

NOTE:

1. PINS 1c, 32a ON EURO CONNECTOR AND PINS Xa, Xc, Ya, Yc, 1c, 32a ON U.S. CONNECTOR SHALL BE +5V.
2. PINS 1a, 3a, 5c, 8a, 10c, 13a, 20c, 23a, 25c, 28a, 30c, 32c SHALL BE CONNECTED TO GROUND
3. PINS 3, 5, 10, 15, 24, 26, 28, 30, 32, 35, 37, 39 SHALL BE CONNECTED TO GROUND.
4. CAPACITORS C3, C10, C15-C28 ARE .01 uf +80%, -20%, 25V

DR BY	E. GRIFFITH	4-7-82
CHK BY		4-19-82
APPD BY		4/21/82
CONTRACT NO.		

ROCKWELL INTERNATIONAL CORPORATION
Electronic Devices Division
4311 Jamboree Road, Newport Beach, CA 92663

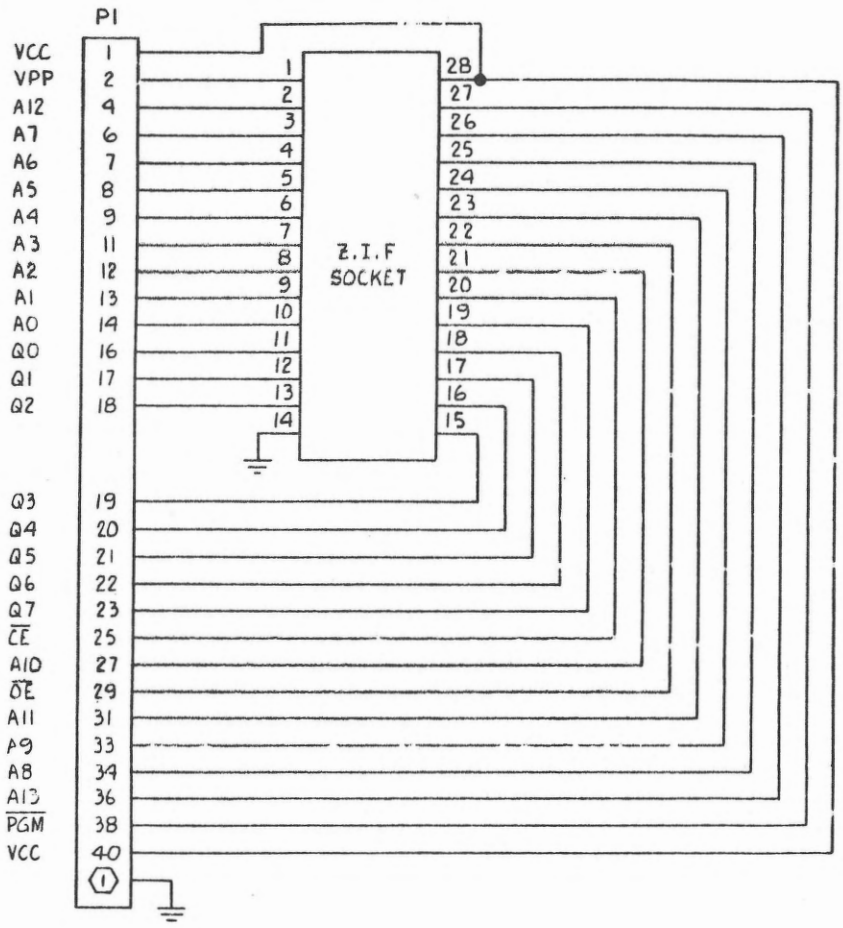
SCHEMATIC- PROM PROGRAMMER RM65

CODE IDENT NO.	DRAWING NO.
34576	PA10-X441

SCALE | SHEET 1 OF 1

PROPRIETARY INFORMATION
Rockwell International Corporation
NO DISSEMINATION OR USE ALLOWED
WITHOUT PRIOR WRITTEN PERMISSION

PRINTED ON KLINGLER 1033-10



NOTES:

(1.) PINS 3,5,10,15,24,26,28,30,32,35,37,39 SHALL BE CONNECTED TO GROUND.

PA10-X444 Schematic PROM Programmer Socket Module

SEMICONDUCTOR PRODUCTS DIVISION REGIONAL ROCKWELL SALES OFFICES

HOME OFFICE

Semiconductor Products Division
Rockwell International
4311 Jamboree Road
P.O Box C, MS 501-300
Newport Beach, California
92658-8902
(714) 833-4700
TWX: 910 591-1698

UNITED STATES

Semiconductor Products Division
Rockwell International
1842 Reynolds
Irvine, California 92714
(714) 833-4655
TWX: 910 595-2518

Semiconductor Products Division
Rockwell International
3375 Scott Blvd., Suite 410
Santa Clara, California 95054
(408) 980-1900
TLX: 756560

Semiconductor Products Division
Rockwell International
2001 N. Collins Blvd., Suite 103
Richardson, Texas 75080
(214) 996-6500
TLX: 501258

Semiconductor Products Division
Rockwell International
10700 West Higgins Rd., Suite 102
Rosemont, Illinois 60018
(312) 297-8862
TWX: 910 233-0179
(RI MED ROSM)

Semiconductor Products Division
Rockwell International
5001B Greentree
Executive Campus, Rt. 73
Marlton, New Jersey 08053
(609) 596-0090
TWX: 710 940-1377

Semiconductor Products Division
Rockwell International
2 Burlington Woods Dr.
Burlington, Massachusetts 01803
(617) 272-5645

FAR EAST

Semiconductor Products Division
Rockwell International
Overseas Corp.
Itohpia Hirakawa-cho Bldg.
7-6, 2-chome, Hirakawa-cho
Chiyoda-ku, Tokyo 102, Japan
(03) 265-8806
TLX: J22198

Rockwell Collins International
Tai Sang Commercial Bldg.,
11th Floor
24-34 Hennessy Rd.
Hong Kong
(5) 274-321
TLX: 74071 HK

EUROPE

Semiconductor Products Division
Rockwell International GmbH
Fraunhoferstrasse 11B
D-8033 Munchen-Martinsried
West Germany
(089) 857-6016
TLX: 0521/2650 rimd d

Semiconductor Products Division
Rockwell International Limited
Heathrow House, Bath Rd.
Cranford, Hounslow,
Middlesex, TW5 9QW England
(01) 759-2366
TLX: 851-25463

Semiconductor Products
Rockwell Collins Italiana S.p.A.
Via Boccaccio, 23
20123 Milano, Italy
(02) 498-7479
TLX: 3185[illegible]

Semiconductor Products Division
Rockwell International
Immeuble de Bureaux Evry 2
523, Place des Terasses
91034 Evry Cedex, France
(6)497.28.28
TLX: 690-328

YOUR LOCAL REPRESENTATIVE

1/85